

RECEIVED  
CENTRAL FAX CENTER  
MAY - 7 2007

HEWLETT-PACKARD COMPANY  
Intellectual Property Administration  
P.O. Box 272400  
Fort Collins, Colorado 80527-2400

## PATENT APPLICATION

ATTORNEY DOCKET NO. 10012176-1

IN THE  
UNITED STATES PATENT AND TRADEMARK OFFICE

Inventor(s): **Damien R. Forkner, et al.**

Confirmation No.: 5672

Application No.: 10/029,774

Examiner: Lilian Vo

Filing Date: December 20, 2001

Group Art Unit: 2105

Title: **PERSISTENT PROCESS SOFTWARE ARCHITECTURE**

Mail Stop Appeal Brief-Patents  
Commissioner For Patents  
PO Box 1450  
Alexandria, VA 22313-1450

**TRANSMITTAL OF APPEAL BRIEF**Transmitted herewith is the Appeal Brief in this application with respect to the Notice of Appeal filed on May 1, 2007.

The fee for filing this Appeal Brief is (37 CFR 1.17(c)) \$500.00.

(complete (a) or (b) as applicable)

The proceedings herein are for a patent application and the provisions of 37 CFR 1.136(a) apply.

☐ (a) Applicant petitions for an extension of time under 37 CFR 1.136 (fees: 37 CFR 1.17(a)-(d)) for the total number of months checked below:

☐ 1st Month  
\$120

☐ 2nd Month  
\$450

☐ 3rd Month  
\$1020

☐ 4th Month  
\$1590

☐ The extension fee has already been filed in this application.

☒ (b) Applicant believes that no extension of time is required. However, this conditional petition is being made to provide for the possibility that applicant has inadvertently overlooked the need for a petition and fee for extension of time.

Please charge to Deposit Account 08-2025 the sum of \$ 500. At any time during the pendency of this application, please charge any fees required or credit any over payment to Deposit Account 08-2025 pursuant to 37 CFR 1.25. Additionally please charge any fees to Deposit Account 08-2025 under 37 CFR 1.16 through 1.21 inclusive, and any other sections in Title 37 of the Code of Federal Regulations that may regulate fees.

☒ A duplicate copy of this transmittal letter is enclosed.

☐ I hereby certify that this correspondence is being deposited with the United States Postal Service as first class mail in an envelope addressed to:  
Commissioner for Patents, Alexandria, VA 22313-1450  
Date of Deposit:

OR

☒ I hereby certify that this paper is being transmitted to the Patent and Trademark Office facsimile number (571)273-8300.

Date of facsimile: May 7, 2007

Typed Name: Douglas L. Weller

Signature: 

Respectfully submitted,

Damien R. Forkner, et al.

By 

Douglas L. Weller

Attorney/Agent for Applicant(s)

Reg No. : 30,506

Date : May 7, 2007

Telephone : (408) 985-0642

Rev 10/06a (Ap/Brief)

RECEIVED  
CENTRAL FAX CENTER

MAY - 7 2007

## HEWLETT PACKARD COMPANY

Intellectual Property Administration  
P.O. Box 272400  
Fort Collins, Colorado 80527-2400

## PATENT APPLICATION

ATTORNEY DOCKET NO. 10012176-1

IN THE  
UNITED STATES PATENT AND TRADEMARK OFFICE

Inventor(s): Damien R. Forkner, et al.

Confirmation No: 5672

Application No: 10/029,774

Examiner: Lilian Vo

Filing Date: December 20, 2001

Group Art Unit: 2195

SUBJECT: PERSISTENT PROCESS SOFTWARE ARCHITECTURE  
-----

## COMMISSIONER FOR PATENTS

P.O. Box 1450  
Alexandria, VA 22313-1450

Sir:

APPEAL BRIEF

Appellant herein sets forth his reasons and arguments for appealing the Examiner's final rejection of claims in the above-identified case.

## REAL PARTY IN INTEREST

This Patent Application has been assigned to Hewlett-Packard Development Company, L.P., a Texas Limited Partnership having its principal place of business in Houston, Texas.

## RELATED APPEALS AND INTERFERENCES

Appellant is aware of no related appeals or interferences.

05/08/2007 EAYALEW1 00000033 002025 10029774  
01 FC:1402 500.00 DA

### STATUS OF CLAIMS

Claims 1 through 4, 6 through 14 and 16 through 21 and 23 through 26 are pending in the case.

Claims 5, 15 and 22 have been canceled.

Claims 1 through 4, 6 through 14 and 16 through 21 and 23 through 26 are rejected.

The appealed claims are claims 1 through 4, 6 through 14 and 16 through 21 and 23 through 26.

### STATUS OF AMENDMENTS

After the final rejection, no amendments to the claims have submitted or entered.

### SUMMARY OF CLAIMED SUBJECT MATTER

#### Independent Claim 1:

Claim 1 sets out a server computing system (26). The server computing system (26) comprises an application (40-44) running on hardware within the server computer system. The application (40-44) comprises a persistent process (40) and a plurality of transient processes (41-44). See Figure 1 and the Specification at page 5 lines 1 through 12.

The persistent process (40) generates dynamic and interactive hypertext markup language (HTML) content for the application (40-44). See the Specification at page 5, lines 9 through 11. Each transient process (41-44) in the

plurality of transient processes (41-44) is launched to handle a client request from a client (21-24). See the Specification at page 5, lines 1 through 8. The client request is handled by parsing the client request, forwarding the client request to the persistent process (40), capturing a result from the persistent process (40) and forwarding the result to the client (21-24). See Figure 2 and the Specification at page 8, lines 1 through 20. The persistent process (40) performs background processing (63) when no client requests are pending. See Figure 3 and the Specification at page 9, lines 9 and 10. The background processing (63) includes caching in memory. See the Specification at page 9, lines 10 and 11 and the Specification at page 5, lines 20 through 21.

**Independent Claim 12:**

Claim 12 sets out a computer implemented method performed within a server (26). See Figure 1 and the Specification at page 5 lines 1 through 2. The method includes running a persistent process (40) that generates dynamic and interactive hypertext markup language (HTML) content for an application (40-44). See the Specification at page 5, lines 9 through 11. The persistent process (40) performs background processing (63) when no client requests are pending. See Figure 3 and the Specification at page 9, lines 9 and 10. The background processing (63) includes caching in memory. See the Specification at page 9, lines 10 and 11 and the Specification at page 5, lines 20 through 21. For each of a plurality of client requests a transient process (41-44) is launched to handle the client request. See the Specification at page 5, lines 1 through 8. Each client

request is parsed by the transient process (41-44). The client request is forwarded to the persistent process (40). A result from the persistent process (40) is captured. The result is forwarded to a client (21-24). See Figure 2 and the Specification at page 8, lines 1 through 20.

**Independent Claim 21:**

Claim 21 sets out storage media that stores a computer application (40-44). The computer application (40-44), when executed on a computing system, comprises a persistent process (40) and a plurality of transient processes (41-44). See Figure 1 and the Specification at page 5 lines 1 through 12. The persistent process (40) generates dynamic and interactive hypertext markup language (HTML) content for the computer application (40-44). See the Specification at page 5, lines 9 through 11. Each transient process (41-44) from the plurality of transient processes (41-44) is launched to handle a client request from a client (21-24). See the Specification at page 5, lines 1 through 8. The client request is handled by parsing the client request, forwarding the client request to the persistent process (40), capturing a result from the persistent process (40) and forwarding the result to the client (21-24). See Figure 2 and the Specification at page 8, lines 1 through 20. The persistent process (40) performs background processing (63) when no client requests are pending. See Figure 3 and the Specification at page 9, lines 9 and 10. The background processing (63) includes caching in memory. See the Specification at page 9, lines 10 and 11 and the Specification at page 5, lines 20 through 21.

## GROUND OF REJECTION TO BE REVIEWED ON APPEAL

(1) Claims 1 through 4, 6 through 14 and 16 through 21 and 23 through 26 stand rejected under 35 U.S.C. § 103 (a) as being unpatentable over USPN 6,237,005 (*Griffin*) in view of USPN 6,243,719 (*Ikuta*).

## ARGUMENT

### 35 U.S.C. § 103 (A) Rejection Over *Griffin* and *Ikuta*

#### A. Overview of Errors in the Rejection

The U.S. Patent and Trademark Office has set forth a methodology for establishing a *prima facie* case of obviousness. Specifically three basic criteria must be met. First, there must be some suggestion or motivation, either in the references themselves or in the knowledge generally available to one of ordinary skill in the art, to modify the reference or to combine reference teachings. Second, there must be a reasonable expectation of success. Finally, the prior art reference (or references when combined) must teach or suggest all the claim limitations. The teaching or suggestion to make the claimed combination and the reasonable expectation of success must both be found in the prior art and not based on applicant's disclosure. *In re Vaeck*, 947 F.2d 488, 20 USPQ2d 1438 (Fed. Cir. 1991). See also MPEP 706.02 (j).

Appellant believes the Examiner has failed to establish a *prima facie* case of obviousness for the claims extant in the present case because there are claim limitations that are not taught or suggested by any of the cited references.

Furthermore, there is no suggestion or motivation, either in the references themselves or in the knowledge generally available to one of ordinary skill in the art, to combine reference teachings as suggested by the Examiner.

## **B. Discussion of independent claim 1**

### 1. Limitations in claim 1 not disclosed or suggested by the cited art

Independent claim 1 sets out a server computing system that includes an application. The application includes a persistent process that generates dynamic and interactive hypertext markup language (HTML) content for the application. The persistent process performs background processing when no client requests are pending. The background processing includes caching in memory. This is not disclosed by *Griffin* or *Ikuta*.

#### a. Discussion of *Griffin*:

*Griffin* discloses a web server mechanism for processing multiple transactions in an interpreted language executed environment. *Griffin* does not disclose or suggest that a persistent process performs background processing when no client requests are pending. Further, *Griffin* does not disclose or suggest that background processing can include caching in memory, as set out in claim 1.

Particularly, *Griffin* does not disclose any persistent process performing background processing when no client requests are pending. In fact, *Griffin* teaches the opposite. For example, Figure 6 of *Griffin* is a flow diagram that

shows functions of the master interpreter of *Griffin*. See column 4, lines 57 through 59. As shown by Figure 6, during an initial state 540, the master interpreter 510 listens for a transaction request message to arrive (step 610). See *Griffin* at column 9, lines 18 through 26. After the transaction, the master interpreter 510 re-enters the initial state 540 and begins to listen for any other transaction request messages (step 610). See *Griffin* at column 9, lines 57 through 60. Thus, *Griffin* teaches that between transactions, master interpreter 510 merely listens for a next transaction request message to arrive. *Griffin* does not disclose or suggest that master interpreter 510 performs background processing when no client requests are pending, as required by claim 1 of the present case.

Additionally, *Griffin* never even brings up the issue of caching. *Griffin* does not disclose or suggest that caching is performed by any process at any time. It is clear therefore that there is no disclosure or suggestion in *Griffin* that master interpreter 510 performs caching in memory as part of background processing when no client requests are pending.

#### b. Ikuta's Discussion of Caching

*Ikuta* discloses a data caching apparatus used for an electronic conference system where an unspecified plurality of users exchange information. See *Ikuta* at column 1, lines 26 through 27. In order to improve response speed of the system when the user of the electronic conference system requests a listing of messages or selects a message, a cache file on the server is used. See *Ikuta* at column 1, lines 40 through 52. Specifically, in *Ikuta*, a cache file 22 of a server 10



stores at least a list of the messages of each of the forums stored in the database 12 as well as a copy of a predetermined number (for example 100) of messages. The cache file 22 is updated in the background when the server 10 has some surplus processing capability. See *Ikuta* at column 6, lines 26 through 32.

c. Discussion of Combination of *Griffin* and *Ikuta*

*Ikuta* discloses storing in a cache file 22 a list of the messages of each of the forums stored in the database 12 as well as a copy of a predetermined number (for example 100) of messages. A person of ordinary skill in the art would have no motivation to add this subject matter into *Griffin*. In *Griffin* there is no discussion of electronic conference systems nor of the need to improve response speed of the system when the user of an electronic conference system requests a listing of messages or selects a message. Therefore, the use of a cache in an electronic conference system such as *Ikuta* would not provide motivation for a person of ordinary skill in the art to modify *Griffin* to include a caching apparatus, as disclosed by *Ikuta*.

i. The Examiner's suggested motivations

The Examiner has made various attempts to show motivation to add background processing and particularly caching in memory to the web server mechanism disclosed in *Griffin*.

For example, the Examiner has argued that *Griffin's* master interpreter "is capable of performing the step of caching in the background when there is no

pending requests so that resource can be fully utilized." See the Office Action dated April 18, 2007 at page 3, lines 5 through 11 and page 6, lines 4 through 10.

The Examiner has also argued: "...a person of an ordinary skill in the art is motivate[d] to add a cache memory to the web server for storing the information of the duration of the master interpreter as it is formed for the particular session. See the Office Action dated April 18, 2007 at page 7, lines 15 through 17.

The Examiner has also asserted that the motivation for the rejection is found in knowledge generally available to one of ordinary skill in the art. See the Office Action dated April 18, 2007 at page 6, lines 17 through 18.

These statements by the Examiner fail to show that there is suggestion or motivation, either in the references themselves or in the knowledge generally available to one of ordinary skill in the art, to modify the *Griffin* as suggested by the Examiner.

#### ii. Discussion of what is meant by motivation

The motivation to make a specific structure is not abstract, but practical, and is always related to the properties or uses one skilled in the art would expect the structure to have, if made. The critical inquiry is whether there is something in the prior art as a whole to suggest the desirability, and thus the obviousness, of making the combination. *In re Newell*, 891 F.2d 899, 13 U.S.P.Q. 2d 1248, 1250 (Fed. Cir. 1989).

The mere fact that a worker in the art could rearrange the parts of the reference device to meet the terms of the claims is not, by itself, sufficient to support a finding of obviousness. The prior art must provide a motivation or reason for the worker in the art, without the benefit of the appellant's specification, to make the necessary changes in the reference device. *Ex parte Chicago Rawhide Manufacturing Co.*, 226 U.S.P.Q 438 (PTO Bd. App. 1984).

iii. Not obvious to add a cache to web server in Griffin

In *Griffin*, the persistent process is a persistent interpreter. There is no teaching or suggestion in *Griffin* of how a cache could be used to any advantage in the interpreting process. Thus, no motivation is provided in *Griffin* for a person of ordinary skill in the art to add a cache memory to the web server mechanism disclosed by *Griffin*.

While *Ikuta* discloses use of a cache file, the cache file is used to store lists of messages and messages within an electronic conference system. Since *Griffin* is not concerned with lists of messages and messages within an electronic conference system, *Ikuta's* teaching about use of a cache to file lists of messages and messages would not motivate a person of ordinary skill in the art to modify *Griffin* to include a cache.

Outside of the desire to perform piecemeal reconstruction of Appellant's claimed subject matter, there is no motivation by which a person of ordinary skill in the art would be motivated to modify *Griffin* to add caching in memory

as part of background processing when no client requests are pending, as set out in claim 1 of the present case.

iv. Response to particular argument made by the Examiner

The Examiner has argued as follows

*Griffin's* master interpreter (persistent process) "...is formed at time 1009 corresponding to a time prior to the receipt of the first transact request message...The duration 1016 of the master interpreter 1002 extends from the time it was formed 1009 to an indefinite time in the future..." (col. 15 lines 41-48)[.] In this case, a person of an ordinary skill in the art is motivate[d] to add a cache memory to the web server for storing the information of the duration of the master interpreter as it is formed for the particular session.

See the Office Action dated April 18, 2007 at page 6, lines 15 through 17.

Here the Examiner appears to be arguing that since master interpreter 1002 is persistent it therefore would be obvious to add a cache memory to the web server disclosed by *Griffin*. This argument by the Examiner is a non sequitur. That is, whether or not a cache memory is used by a web server is unrelated to whether an interpreter is a persistent process.

There is nothing in *Griffin* to suggest that master interpreter 1002 would have any use for a cache memory. The fact that master interpreter 1002 is persistent does not disclose or suggest that a cache memory should be added to the web server disclosed by *Griffin*. A person of ordinary skill in the art would see no connection between the fact that master interpreter 1002 is persistent and the decision as to whether a cache memory should be added to the web server disclosed by *Griffin*.

Nothing in *Griffin* discloses or suggests the benefit of adding a cache memory to the web server in *Griffin*. Nothing in *Ikuta* discloses or suggests any benefit to adding a cache memory to the web server in *Griffin*. Therefore, a person of ordinary skill in the art would not find any motivation in *Ikuta* or *Griffin* to add a cache memory to the web server in *Griffin*.

v. Generally available knowledge

The Examiner has asserted that the motivation for the rejection is found in knowledge generally available to one of ordinary skill in the art. See the Office Action dated April 18, 2007 at page 6, lines 17 through 18. However, the Examiner has given no further explanation or documentation of what is meant by this.

Appellant notes that this line of argument was first raised by the Examiner in the Final Office Action dated April 18, 2007. This Appeal Brief is therefore the first opportunity Appellant has had to respond to this argument.

Appellant respectfully traverses the Examiner's assertion that the motivation for the rejection is found in knowledge generally available to one of ordinary skill in the art.

Official notice unsupported by documentary evidence should only be taken by the examiner where the facts asserted to be well-known, or to be common knowledge in the art are capable of instant and unquestionable demonstration as being well known. As noted by the court in *In re Ahlert*, 424 F.2d 1088, 1091, 165 USPQ 418, 420 (CCPA 1970), the notice of facts beyond the

record which may be taken by the examiner must be "capable of such instant and unquestionable demonstration as to defy dispute" (citing *In re Knapp Monarch Co.*, 296 F.2d 230, 132 USPQ 6 (CCPA 1961)).

Appellant does not believe that motivation to combine *Griffin* and *Ikuta* as suggested by the Examiner is capable of instant and unquestionable demonstration as being well known. Appellant therefore requests from the Examiner documentation to substantiate this argument.

### C. Discussion of independent 12

#### 1. Limitations in claim 12 not disclosed or suggested by the cited art

Independent claim 12 sets out a computer-implemented method. In step (a) of claim 12, a persistent process that generates dynamic and interactive hypertext markup language (HTML) content for an application is run. The persistent process performs background processing when no client requests are pending. The background processing includes caching in memory. This is not disclosed by *Griffin* or *Ikuta*.

As discussed above, *Griffin* discloses a web server mechanism for processing multiple transactions in an interpreted language executed environment. *Griffin* does not disclose or suggest that a persistent process performs background processing when no client requests are pending, where the background processing includes caching in memory, as set out in claim 12.

Particularly, *Griffin* does not disclose any persistent process performing background processing when no client requests are pending. In fact, *Griffin*

teaches the opposite. For example, Figure 6 of *Griffin* is a flow diagram that shows functions of the master interpreter of *Griffin*. See column 4, lines 57 through 59.

As shown by Figure 6, during an initial state 540, the master interpreter 510 listens for a transaction request message to arrive (step 610). See *Griffin* at column 9, lines 18 through 26. After the transaction, the master interpreter 510 re-enters the initial state 540 and begins to listen for any other transaction request messages (step 610). See *Griffin* at column 9, lines 57 through 60. Thus, *Griffin* teaches that between transactions, master interpreter 510 merely listens for a next transaction request message to arrive. *Griffin* does not disclose or suggest that master interpreter 510 performs background processing when no client requests are pending, where the background processing includes caching in memory, as required by claim 12 of the present case.

Additionally, *Griffin* never even brings up the issue of caching. *Griffin* does not disclose or suggest that caching is performed by any process at any time. It is clear therefore that there is no disclosure or suggestion in *Griffin* that master interpreter 510 performs caching in memory as part of background processing when no client requests are pending.

## 2. Combination of *Griffin* and *Ikuta*

*Ikuta* discloses storing in a cache file 22 a list of the messages of each of the forums stored in the database 12 as well as a copy of a predetermined number (for example 100) of messages. A person of ordinary skill in the art would have

no motivation to add this subject matter into *Griffin*. In *Griffin* there is no discussion of electronic conference systems nor of the need to improve response speed of the system when the user of the electronic conference system requests a listing of messages or selects a message. Therefore, the use of a cache in an electronic conference system such as *Ikuta* would not provide motivation for a person of ordinary skill in the art to modify *Griffin* to include a caching apparatus, as disclosed by *Ikuta*.

In *Griffin*, the persistent process is a persistent interpreter. There is no teaching or suggestion in *Griffin* of how a cache could be used to any advantage in the interpreting process. Thus, no motivation is provided in *Griffin* or *Ikuta* for a person of ordinary skill in the art to add a cache memory to the web server mechanism disclosed by *Griffin*.

While *Ikuta* discloses use of a cache file, the cache file is used to store lists of messages and messages within an electronic conference system. Since *Griffin* is not concerned with lists of messages and messages within an electronic conference system, *Ikuta*'s teaching about use of a cache file lists of messages and messages would not motivate a person of ordinary skill in the art to modify *Griffin* to include a cache.

#### D. Discussion of independent 21

##### 1. Limitations in claim 21 not disclosed or suggested by the cited art

Independent claim 21 sets out storage media that stores a computer application. The computer application, when executed on a computing system



comprises a persistent process that generates dynamic and interactive hypertext markup language (HTML) content for the computer application. The persistent process performs background processing when no client requests are pending, the background processing including caching in memory. This is not disclosed by *Griffin* or *Ikuta*.

As discussed above, *Griffin* discloses a web server mechanism for processing multiple transactions in an interpreted language executed environment. *Griffin* does not disclose or suggest that a persistent process performs background processing when no client requests are pending, where the background processing includes caching in memory, as set out in claim 21.

Particularly, *Griffin* does not disclose any persistent process performing background processing when no client requests are pending. In fact, *Griffin* teaches the opposite. For example, Figure 6 of *Griffin* is a flow diagram that shows functions of the master interpreter of *Griffin*. See column 4, lines 57 through 59.

As shown by Figure 6, during an initial state 540, the master interpreter 510 listens for a transaction request message to arrive (step 610). See *Griffin* at column 9, lines 18 through 26. After the transaction, the master interpreter 510 re-enters the initial state 540 and begins to listen for any other transaction request messages (step 610). See *Griffin* at column 9, lines 57 through 60. Thus, *Griffin* teaches that between transactions, master interpreter 510 merely listens for a next transaction request message to arrive. *Griffin* does not disclose or suggest that master interpreter 510 performs background processing when no

client requests are pending, where the background processing includes caching in memory, as required by claim 21 of the present case.

Additionally, *Griffin* never even brings up the issue of caching. *Griffin* does not disclose or suggest that caching is performed by any process at any time. It is clear therefore that there is no disclosure or suggestion in *Griffin* that master interpreter 510 performs caching in memory as part of background processing when no client requests are pending.

## 2. Combination of *Griffin* and *Ikuta*

*Ikuta* discloses storing in a cache file 22 a list of the messages of each of the forums stored in the database 12 as well as a copy of a predetermined number (for example 100) of messages. A person of ordinary skill in the art would have no motivation to add this subject matter into *Griffin*. In *Griffin* there is no discussion of electronic conference systems nor of the need to improve response speed of the system when the user of the electronic conference system requests a listing of messages or selects a message. Therefore, the use of a cache in an electronic conference system such as *Ikuta* would not provide motivation for a person of ordinary skill in the art to modify *Griffin* to include a caching apparatus, as disclosed by *Ikuta*.

In *Griffin*, the persistent process is a persistent interpreter. There is no teaching or suggestion in *Griffin* of how a cache could be used to any advantage in the interpreting process. Thus, no motivation is provided in *Griffin* or *Ikuta*

for a person of ordinary skill in the art to add a cache memory to the web server mechanism disclosed by *Griffin*.

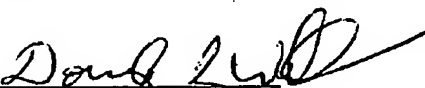
While *Ikuta* discloses use of a cache file, the cache file is used to store lists of messages and messages within an electronic conference system. Since *Griffin* is not concerned with lists of messages and messages within an electronic conference system, *Ikuta*'s teaching about use of a cache file lists of messages and messages would not motivate a person of ordinary skill in the art to modify *Griffin* to include a cache.

#### CONCLUSION

For all the reasons discussed above, Appellant believes the rejection of the claims was in error and respectfully requests that the rejection be reversed.

Respectfully submitted,

DAMIEN R. FORKNER, ET AL.

By   
Douglas L. Weller  
Reg. No. 30,506

May 7, 2007  
Santa Clara, California  
(408) 985-0642

### CLAIMS APPENDIX

1. (Previously Presented) A server computing system comprising:  
an application running on hardware within the server computer system,  
the application comprising:  
a persistent process that generates dynamic and interactive  
hypertext markup language (HTML) content for the application; and,  
a plurality of transient processes, wherein each transient process is  
launched to handle a client request from a client by parsing the client request,  
forwarding the client request to the persistent process, capturing a result from  
the persistent process and forwarding the result to the client;  
wherein the persistent process performs background processing when no  
client requests are pending, the background processing including caching in  
memory.
2. (Previously Presented) A server computing system as in claim 1  
wherein the persistent process utilizes a support process outside the server.
3. (Previously Presented) A server computing system as in claim 1  
wherein the transient processes implement a Common Gateway Interface (CGI).
4. (Previously Presented) A server computing system as in claim 1  
wherein the persistent process includes a request queue.

5. (Canceled)

6. (Previously Presented) A server computing system as in claim 1 wherein each of the plurality of transient processes terminates after forwarding the result to the client.

7. (Previously Presented) A server computing system as in claim 1 wherein when a first client sends a file request for a file, a first transient process obtains and forwards the file to the first client.

8. (Previously Presented) A server computing system as in claim 1 wherein when a first client sends a file request for a file, a first transient process, after verifying access to the file, obtains and forwards the file to the first client.

9. (Previously Presented) A server computing system as in claim 1 wherein the plurality of transient processes communicate with the persistent process via Interprocess Communication (IPC).

10. (Previously Presented) A server computing system as in claim 1 wherein the caching in memory performed by the persistent process is look-ahead caching.

11. (Previously Presented) A server computing system as in claim 1 wherein the persistent process uses a queue to process client requests forwarded by the plurality of transient processes to the persistent process.

12. (Previously Presented) A computer implemented method performed within a server, the method comprising the following:

(a) running a persistent process that generates dynamic and interactive hypertext markup language (HTML) content for an application, the persistent process performing background processing when no client requests are pending, the background processing including caching in memory; and,

(b) for each of a plurality of client requests, performing the following:

(b.1) launching a transient process to handle each client request,

(b.2) parsing each client request by the transient process,

(b.3) forwarding the client request to the persistent process,

(b.4) capturing a result from the persistent process, and

(b.5) forwarding the result to a client.

13. (Previously Presented) A computer implemented method as in claim 12 wherein (a) includes the following:

(a.1) utilizing, by the persistent process, a support process outside the server.

14. (Previously Presented) A computer implemented method as in claim 12 wherein the transient processes implement a Common Gateway Interface (CGI).

15. (Canceled)

16. (Previously Presented) A computer implemented method as in claim 12 wherein (b) additionally includes the following:

(b.6) terminating the transient process after forwarding the result to the client.

17. (Previously Presented) A computer implemented method as in claim 12 additionally comprising the following:

(c) when a first client sends a file request for a file, performing the following:

(c.1) obtaining, by a first transient process, the file, and

(c.2) forwarding, by the first transient process, the file to the first client.

18. (Previously Presented) A computer implemented method as in claim 12 additionally comprising the following:

(c) when a first client sends a file request for a file, performing the following:

(c.1) verifying a right of the first client to access the file,  
(c.2) obtaining, by a first transient process, the file, and  
(c.3) forwarding, by the first transient process, the file to the first client.

19. (Previously Presented) A computer implemented method as in claim 12 wherein the caching in memory performed by the persistent process is look-ahead caching.

20. (Previously Presented) A computer implemented method as in claim 12 wherein (a) includes the following:

(a.1) using a queue to process client requests forwarded to the persistent process.

21. (Previously Presented) Storage media that stores a computer application, the computer application, when executed on a computing system, comprising:

a persistent process that generates dynamic and interactive hypertext markup language (HTML) content for the computer application; and,  
a plurality of transient processes, wherein each transient process is launched to handle a client request from a client by parsing the client request, forwarding the client request to the persistent process, capturing a result from the persistent process and forwarding the result to the client;



wherein the persistent process performs background processing when no client requests are pending, the background processing including caching in memory.

22. (Canceled)

23. (Original) Storage media as in claim 21 wherein each of the plurality of transient processes terminates after forwarding the result to the client.

24. (Original) Storage media as in claim 21 wherein when a first client sends a file request for a file, a first transient process obtains and forwards the file to the first client.

25. (Previously Presented) Storage media as in claim 21 wherein the caching in memory performed by the persistent process is look-ahead caching.

26. (Original) Storage media as in claim 21 wherein the persistent process uses a queue to process client requests forwarded by the plurality of transient processes to the persistent process.

#### **EVIDENCE APPENDIX**

No evidence under §§ 1.130, 1.131, or 1.132 is relied upon by Appellant in the appeal.

#### **RELATED PROCEEDINGS APPENDIX**

There are no related decisions rendered by a court or the Board.